

METHOD, SYSTEM, AND PROGRAM FOR PREPROCESSING
A DOCUMENT TO RENDER ON AN OUTPUT DEVICE

BACKGROUND OF THE INVENTION

5 1. Field of the Invention

The present invention relates to a method, system, and program for Preprocessing a document for rendering on an output device.

2. Description of the Related Art

10 Many documents and data objects are encoded in the Extensible Mark-up Language (XML) as structured documents. Numerous categories of data can be encoded within an XML structured document, such as vector graphics, e-commerce transactions, mathematical equations, object meta-data, server APIs, and a thousand other kinds of structured information. XML documents include tags to mark a start and end of each of the logical parts (called
15 elements) of the document. For instance, if the XML document defines a book, the elements would include the table of contents, chapters, paragraphs, appendices, etc. Each element may include content and one or more attributes. An XML document further includes a definition of each element in a formal model, known as a Document Type Definition (DTD). The DTD provides attributes for each element and indicates the relationship of the elements and possible
20 attribute values. Elements may be arranged in a hierarchical relationship. In such case, the DTD would define the hierarchical relationship of the elements to one another. Further details of XML are described in the publication "Extensible Markup Language (XML) 1.0", Second Edition (Copyright W3C, Oct. 6, 2000), which publication is incorporated herein by reference in its entirety.

25 The content of an XML document may be printed using the Extensible Style Language (XSL). XSL includes both a transformation language and a formatting language. An XSL

stylesheet processor accepts a document or data in XML and an XSL stylesheet and produces the presentation of that XML source content that was intended by the designer of that stylesheet. There are two aspects of this presentation process: first, constructing a result tree from the XML source tree and second, interpreting the result tree to produce formatted results suitable for presentation on a display, on paper, in speech, or onto other media. The first aspect is called tree transformation and the second is called formatting.

Both the source tree and result tree may be expressed as a Document Object Model tree of nodes for the elements of the XML source document. Tree transformation allows the structure of the result tree to be significantly different from the structure of the source tree. For example, one could add a table-of-contents as a filtered selection of an original source document, or one could rearrange source data into a sorted tabular presentation. In constructing the result tree, the tree transformation process also adds the information necessary to format that result tree for presentation. Formatting is enabled by including formatting semantics in the result tree. Formatting semantics are expressed in terms of a catalog of classes of formatting objects. The nodes of the result tree are formatting objects. The classes of formatting objects denote typographic abstractions such as page, paragraph, table, font, margins, etc. Finer control over the presentation of these abstractions is provided by a set of formatting properties, such as those controlling indents, word- and letter-spacing, and widow, orphan, and hyphenation control. In XSL, the classes of formatting objects and formatting properties comprise a presentation language for expressing presentation intent.

The output of the XSL transformation may comprise an HTML document to render within a web browser program, e.g., Microsoft Internet Explorer, Netscape Communicator, etc., or text, graphics or image output for rendering on a printer device. Current techniques exist for transforming an XML document to output that can be transformed or rasterized, such as a page description language (PDL), e.g., PostScript, Portable Document Format (PDF), etc.** To transform the XML document, the entire source file must be

rendered in memory and then transformed to a result tree including formatting objects that define an internal representation of the page layouts. Details of formatting such as page size, element size, font, color, margin, page dimensions, etc., are specified by properties. Formatting properties are represented as attributes on the individual formatting object elements. A special

5 XSL application, referred to as an XSL-FO (XSL-Formatting Objects) application, generates from the source XML document a result XML tree including formatting objects to define the page layout for presentation to a reader.

In the prior art, there are application programs that convert an XML-FO document into a viewable format such as PDF or TeX. For instance, the Apache XML Project

10 provides a Java application, known as "FOP", that reads an XML formatting object tree and then turns it into a PDF document. The Java application must use an XML parser, such as the Simple API for XML (SAX), to parse the XML-FO result tree and then transform the tree elements into output in a PDL format, such as PDF or PostScript. Alternatively, the Result document including formatting objects can be passed into memory as a DOM document. In

15 such case, the Java application would use DOM commands to access the elements of the result tree in the DOM document.

Other prior art techniques for outputting the text into a PDL format involves first transforming the result XML document into a output in a TeX format, which may then be transformed into PostScript or PDF.

20 With such prior art methods, the generated PDL output is device dependent and optimized for rendering on a specific printer device. One drawback with device dependent output is that it does not produce optimal results when rendered on other devices or rendered on the specific device at a later time. To provide optimal output, the output device would have to itself transform the entire XML document in memory to a result document, then transform the

25 result document into a printer transformable output, such as a page description language (PDL), and then finally rasterized into printer ready output. Such prior art XSL techniques for

transforming the XML document require substantial processor and memory resources to render both the source document, result document, PDL data stream, and rasterized data

For these reasons, there is a need in the art for improved techniques for transforming XML documents into output for a printer.

5

SUMMARY OF THE PREFERRED EMBODIMENTS

Provided is a method, program and system for processing a source document in a structured document format including elements providing content to render. A source document and page layout data structure providing formatting properties specifying a layout and
10 format of the content output are received. The source document and the page layout data structure are processed to determine page divisions and formatting properties for the content in the source document. Multiple page objects are generated, wherein each page object includes content and formatting properties for at least one page. The page objects are transmitted to a rasterizer to transform into renderable information capable of being generated by an output
15 device.

In further implementations, the source document includes content in a first presentation language. The source document is transformed into a result document in a second presentation language. The result document includes the content from the source document and the formatting properties provided by the page layout data structure. The formatting properties
20 indicate page divisions of the content. The multiple page objects are generated from the result document.

Still further, the first presentation language may comprise the Extensible Markup Language (XML), the second presentation language comprises the Extensible Stylesheet Language Formatting Objects (XSL-FO), and the page layout data structure comprises an
25 XSL stylesheet.

Additionally, the page objects may include content and formatting properties in a device independent presentation language.

The described implementations provide a technique of preprocessing a source document that relieves the output device that renders the source document into renderable information from processing burdens by performing a substantial amount of the preprocessing prior to rendering at the output device. In certain implementations, the source document is transformed to page objects in a device independent format that provides for optimal rendering at the output device that may be rendered by the output device on a page-by-page basis without having to transform the entire source document.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 is an illustration of an implementation of a computing environment in which certain of the described implementations are embodied;

FIG. 2 illustrates an arrangement of a document including Extensible Stylesheet Language Formatting Objects (XSL-FO); and

FIG. 3 illustrates logic to preprocess source documents in accordance with certain implementations.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments of the present invention. It is understood that other embodiments may be utilized and structural and operational changes may be made without departing from the scope of the present invention.

FIG. 1 illustrates a computing environment in which the invention may be implemented. A printing system 2 includes an XML preprocessor 4, a printer driver 6, and a printer 8. The printing system 2 includes one or more computer systems in which the XML preprocessor 4 and printer driver 6 programs are executed. For instance, the XML preprocessor 4 and printer driver 6 may be located on the same computer system, such as a printer server that provides data to the printer 8 over a network to render. Alternatively, the XML preprocessor 4 and the printer driver 6 may execute on separate computers that communicate over a network. The printing system 2 includes communication interfaces 10a, b to provide communication between the XML preprocessor 4, printer driver 6, and printer 8. If the XML preprocessor 4, printer driver 6, and printer 8 are implemented in separate computing devices in a network, then the communication interfaces 10a, b would comprise network interfaces to a network, such as a Local Area Network (LAN), Intranet, the Internet, etc. Alternatively, if the XML preprocessor 4 and printer driver 6 are implemented within the same computer device, then the communication interface 10a, may comprise a bus interface within a computer device. Similarly, if the printer driver 6 and printer 8 are implemented on the same computer device, then the communication interface 10a, may comprise a bus interface within a computer device or output device, such as a printer. Additionally, if the preprocessor 4, printer driver 6, and printer 8 are implemented on the same computer device, then the communication interfaces 10a, b may comprise one or more bus interfaces within the same computer device, such as a printer with significant computational capabilities.

The XML preprocessor 4 includes an XSL processor 12 program and a page separator program 14. The XSL processor 12 is capable of receiving as input a source document 16 in the XML presentation language and an XSL style sheet 18 and transform the XML source document 16 to an Result document 20 including XSL formatting objects (XSL-FO) in a manner known in the art. FIG. 2 illustrates certain of the formatting object components that may be included in the Result document 20. The formatting objects include

formatting properties. The term "formatting properties" as used herein describes any information used to express the layout and presentation of the accompanying content, such as page layout, fonts, page size, element size, color, margins, headers, static information, page numbering, indents, word-and letter-spacing, widow and orphan setting, hyphenation and any
5 other format feature known in the art used to define the appearance of a page of content.

As shown in FIG. 2, the result document 20 includes a root element 50, which declares the namespace and function as the root of the document 20. The root element 50 includes one layout-master-set element 52 and one or more page sequence elements 54. The page-
sequence elements 54 contain content, such as text and images to place on the pages. The
10 page-sequence element may contain text for one or more pages as defined according to the specified simple-page master element defining the layout of output on the pages. The layout-master-set 52 includes one or more templates, referred to as simple-page-masters 56, that contains templates for the pages that will be created. The page-sequence element 54 specifies the name of a template, or simple-page master 56, to use to provide the layout of the content
15 on the page. Each simple-page master 56 defines a general layout for a page including its margins, the sizes of the header, footer, and body area of the page, etc. Each actual page in the rendered document is based on the content and simple-page-master specified in the page-sequence elements 54. Further details of XSL formatting objects are described in the publication entitled "XML Bible", at Chapter 15, by Elliotte Rusty Harold (Copyright 1999),
20 which publication is incorporated herein by reference in its entirety.

The page separator program 14 receives as input the Result document 20 including the XSL formatting objects and generates one or more page objects 22a....n including the content and layout information for each page in the Result document 20. In implementations where the Result document 20 includes the content and formatting properties using the XSL formatting
25 object presentation language, each page object 22a....n would include the content and layout specified for each page-sequence element 54 in the Result document 20. As mentioned, the

formatting properties for a page are provided in the simple-page master specified in the page-sequence element. Further, the content included in the page object 22a....n includes the coding in the language of the original source document 16 that is used to render the content output, such as XML code, PostScript commands, ASCII text, JPEG file, etc. Later, the content in
5 the page object 22a....n is transformed to output that the printer or some other device rasterizes to produce output.

In certain implementations, the page separator program 14 may create a page object 22a....n in a presentation language that describes the formatting properties and content that is different from XML or XSL formatting objects. A presentation language provides constructs
10 to use to specify the formatting properties to present the content. The document described with the presentation language can be interchanged with different application programs and output devices such that the presentation document produces the same document content in the same format on different printers or display devices dependent, however, on the capabilities of each of the printers or display devices.

15 For instance, the page separator program 14 may transform the result document 20 including XSL formatting objects into page objects 22a....n in a presentation language that conforms to the International Business Machine Corporation's (IBM) Mixed Object Document Content Architecture (MO:DCA), which is described in the publication entitled "MO:DCA Reference", IBM document no. SC31-6802-04 (IBM Copyright, 1997), which IBM
20 publication is incorporated herein by reference in its entirety.** In the MO:DCA specification, some of the formatting properties for the object are included in environment information that provides information on the logical page, including measurement units, page width, page depth, and page resource requirements. In MO:DCA implementations, each page object 22a....n would comprise one MO:DCA page. The page layout and formatting provided in the simple-
25 page master XSL formatting object for the page may be encoded into an active environment group in the MO:DCA page object 22a....n. In this way, the formatting specifications of a

simple-page-master in the XSL formatting object language are mapped or transformed into the MO:DCA environment groups of a page object 22a....n. In certain implementations, certain of the layout and formatting information may not map to the active environment group. That layout and formatting information not mapped to the active environment group may be included in the
5 page content, such as the XML code.

The generated page objects 22a....n, including the formatting properties specified in the XSL stylesheet 18 used to produce the Result document 20 and content in the original XML source document 16 code, are then provided to the printer driver 6. The printer driver 6 is a software component that uses a print communication protocol to manage the transfer of pages
10 to a printer interface 24 in the printer 8, and provides for error recovery and other page transfer management operations in a manner known in the art. For instance, the printer driver 6 may utilize the the IBM Intelligent Printer Data Stream (IPDS) printer service that converts a data stream to an IPDS compatible format.** This IPDS data stream is then sent to the printer interface 24 to rasterize and output. The IPDS architecture provides specific commands and
15 status requests. The printer driver 6 may use the IPDS commands to control how the printer 8 processes and rasterizes the page objects 22a....n and monitors printer operations. The IPDS architecture defines bidirectional command protocols for query, resource management, and error recovery. Details of the IPDS architecture and acknowledgment protocol are described in the IBM publication "Intelligent Printer Data Stream Reference", IBM document no. S544-
20 3417-05 (Copyright IBM, 1996), which publication is incorporated herein by reference in its entirety.

The printer driver 6 communicates the pages to a printer interface 24, which may be implemented as a software program executed by the printer 8 processor that is capable of supporting the printer communication protocol used by the printer driver 6, such as the IPDS
25 printer protocol. The printer interface 24 then calls the rasterizer 26 to transform the content in the page object 22a....n into raster data according to formatting information and properties

included in the page object 22a....*n*. In other words, the rasterizer 26 transforms the presentation language used to express the page object 22a....*n* into raster data capable of being rendered by the printer mechanisms 28. The rasterizer 26 may be implemented as software executed by a printer 8 processor within printer 8 memory, e.g., RAM. Alternatively, the rasterizer 26 may comprise hardware rasterizers implemented in a dedicated integrated circuit, such as an application specific integrated circuit (ASIC), Field Programmable Gate Array (FPGA), etc. There may be multiple rasterizers 26 to rasterize pages objects 22a....*n* in parallel. The output of the rasterizer 26 is sent to a printer mechanism 28, which includes the rollers as well as ink dispenser, e.g., laser jet, ink jet, etc.

10 FIG. 3 illustrates logic implemented in the XML pre-processor 4 to process an XML source document 16 and generate the page objects 22a....*n*. Control begins at block 100 with the XML preprocessor 4 receiving an XML source document 16. The XSL processor 12 is called (at block 102) and provided the XSL stylesheet 18 specified for the XML source document 16. The XSL processor 12 transforms the XML code in the source document 16
15 into the result document 20 XML source content and XSL formatting objects (XSL-FO) according to the page layout and formatting properties specified in the XSL stylesheet in a manner known in the art. The XML preprocessor 4 receives the result document 20 outputted from the XSL processor 12 and calls (at block 106) the page separator program 14 to generate the page objects 22a....*n* from the Result document 20. The page separator program
20 14 would include a parser to process and read the XSL formatting object statements in the result document 20. Blocks 108-130 illustrate operations implemented in the page separator program 14 to transform the result document 20 into page objects 22a....*n*.

At block 108, the page separator program 14 receives the result document 20. For each page-sequence element *i* in the result document 20, the page separator program 14
25 accesses (at block 112) the first content element in the page sequence element *i*. The page separator program 14 further processes (at block 114) the simple-page master element

specified in the page sequence element i to determine the formatting properties for the page. The formatting properties included in the simple-page-master are then mapped into the page object 22a... n being constructed. For instance, in implementations where the page object 22a... n is expressed in the MO:DCA presentation language, some of the formatting properties included in the simple-page-master, which is expressed in the XSL-FO presentation language, are mapped to environment groups in the page object that contain formatting information for the page. In such MO:DCA implementations, the page separator program 14 would include code to transform XSL formatting properties in the simple-page master to formatting properties in the MO:DCA language. A new page object is created (at block 116) and the determined page layout is mapped (at block 118) into page layout constructs in the page object. The page separator program 14 inserts (at block 120) the accessed content element into the page object.

If (at block 122) there is another content element in page-sequence element i , then the next content element in page-sequence element i is accessed (at block 124). If (at block 126) there is room in the page object for the accessed next content element, then control returns to block 120 to insert the accessed content element into the page object. Otherwise, if the page object does not include enough space for the next content element, the page separator program 14 outputs (at block 128) the page object and proceeds back to block 116 to create another page object for the further content element(s) in page sequence element i . If (at block 122) there is not another content element in page-sequence element i , then control proceeds (at block 130) to return to block 110 to process the next page-sequence element i , if there are further page sequence elements.

The outputted page objects 22a... n may be stored or archived for later printing. To print the page objects 22a... n , the page objects 22a... n for an XML source document 16 are sent to the printer driver 6 to, in turn, send to the printer interface 24 in the printer 8. The printer interface 24 would call the rasterizer 26 to transform the page objects 22a... n into printer ready raster data that can be rendered by the printer mechanisms 28.

In certain implementations, the page separator program 14 may completely transform the XSL formatted objects in the result document 20 into a data stream conforming to a standard architecture used by applications to describe documents and object envelopes for interchange with other applications and application services, such as the MO:DCA architecture.

- 5 In alternative implementations, the page objects 22a...n may be expressed in other presentation languages, including page description language (PDL) presentation languages, e.g., PDF, PostScript, etc. Page objects in a standard presentation document format may be rasterized by standard transforms known in the art.

- 10 In the described implementations, the page separator program 14 may segregate the result document 20 into page objects 22a...n that include XSL-FO presentation language statements. In such case, the rasterizer 26 in the printer 8 would include a parser and code to transform the XSL formatted objects in the page objects 22a....n into print ready raster data. Additionally, the printer rasterizer 26 may convert the XSL formatted objects in the page objects 22a....n into a device independent page description language (PDL) data stream and
15 then perform a further transform from the PDL data stream to printer ready raster data in a manner known in the art.

- In the above described implementations, the page objects 22a...n include the page content in a device independent presentation language, such as PostScript, PDF, XML formatted objects, etc. This allows the page objects 22a...n to be stored or archived and then
20 later rendered on different output devices in a manner that provides optimal output as the particular device rasterizes the page objects 22a...n according to its device specific rasterizer. Moreover, the described implementations improve the processing speed of the XML source document 16 because the XML preprocessor 4 performs a substantial amount of the transformation operations up front in the page objects 22a...n. In this way, the XML
25 preprocessor 4 relieves the printer 8 of substantial processing burdens during the rasterizing process. Still further, by dividing the result document 20 into multiple page objects 22a....n, the

printer driver 6 may use print facility services known in the art, such as IPDS services, to manage the transfer of the page objects 22a... n to the printer interface 24. Yet further, the printer 8 may rasterize and output the page objects 22a... n for an XML source document 16 on a page-by-page basis, instead of having to render and transform the entire XML source document 16 in memory before sending to the output device to render.

Following are some further implementations.

The XML pre-processor 4, printer driver 6, printer interface 24, and rasterizer 26, and any other computational components, may be implemented as a method, apparatus or program using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The program, code and instructions in which the preferred embodiments are implemented are accessible from and embedded in an information bearing medium, which may comprise one or more computer-readable devices, firmware, programmable logic, memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, SRAMs, etc.), hardware, electronic devices, a computer readable non-volatile storage unit (e.g., CD-ROM, floppy disk, hard disk drive, etc.), a file server providing access to the programs via a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Of course, those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention.

In described implementations, the rasterized data is sent to a printer mechanism 28 to generate print output. However, in alternative embodiments the raster objects may provide output data for another device, such as a display monitor, a storage device for future rendering, etc.

In described implementations, the printer driver 6 sends pages of data to the printer 8. In alternative embodiments, other units of data may be sent, such as objects, chapters, commands, etc., instead of pages.

In described implementations, the printer driver and printer interface are located on separate machines. In alternative embodiments, the printer driver and printer interface may be located on the same machine.

In the described implementations, the format and layout of the content of the page was provided to the XSL processor through an XSL stylesheet. In alternative implementations, the source document may be transformed without using an external stylesheet to provide layout information. For instance, the elements within the source document may include internal layout information.

Each processing component, such as the XSL processor 12, page separator program 14, printer driver 6, printer interface 24, and rasterizer 26 may execute on one or more computer systems. Multiple processing components may be implemented on the same computer system or distributed across multiple computer systems.

In the described implementations, the XML source document was in XML and the transformed result document included XSL formatting objects. In additional implementations, the source document and result document may be expressed in a different presentation language other than XML or XSL formatting objects. For instance, the source document presentation language may comprise another structured document language, such as Dynamic Hypertext Mark-Up Language (DHTML), the Extensible Markup Language (XML), Cascading Style Sheets, any other Standard Generalized Markup Language (SGML), or any other language known in the art for creating interchangeable, structured documents. In yet further embodiments, the requested file may be in any other file format, i.e., other than an SGML type format, capable of being displayed or otherwise executed by the requesting client. Still further, the source document may be written in a page description language, such as PostScript, PDF, etc., wherein the individual page object would include content in the page description language to print on separate pages.

In the described implementations, the page objects 22a...n utilize a device independent presentation language describing the formatting properties that is rasterized by the printer 8 before printing. In alternative implementations, the page objects 22a...n may be rasterized before transmission to the printer 8 or other output device. Such implementations would further
5 relieve the printer 8 of processing burdens.

The program flow logic described in the flowcharts indicated certain events occurring in a certain order. Those skilled in the art will recognize that the ordering of certain programming steps or program flow may be modified without affecting the overall operation performed by the preferred embodiment logic, and such modifications are in accordance with the preferred
10 embodiments.

In the described implementations, the elements in the documents as arranged in a hierarchical fashion. However, in alternative embodiments, the elements do not have to be in a hierarchical relationship to one another.

In the described implementations, each page object 22a...n included information to
15 render one page of output. In alternative printing environment implementations, each page object 22a...n may include content and formatting information to render multiple pages of output.

The XML source document 16 may maintain instances of any type of data object that can be expressed as a structured document, such as print jobs, newspaper articles, Internet
20 Web pages, books, etc.

The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited
25 not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of

the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

5

**PostScript is a registered trademark of Adobe Systems, Inc.; IPDS and MO:DCA are trademarks of International Business Machines Corporation.

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196